Squidbox Code Documentation - Cole Gilbert & George Shui

Git Repo: https://github.com/EAMIRorg/Squidbox

We made the following key changes:

1. **Added Randomized Velocity Logic**

We modified how MIDI notes are sent so that velocity is no longer always fixed at 127. Instead, the system can now generate a random velocity within a defined range.

2. **Centralized Velocity Control in Squidbox**

We added new variables and helper functions in the Squidbox class to manage velocity behavior globally:

- **randomVelocityEnabled** (boolean toggle)
- **fixedVelocity**
- **randomVelocityMin**
- **randomVelocityMax**

src > Squidbox.cpp > ...

```
122  //All helper functions for random velocity below
123
124  bool Squidbox::isRandomVelocityEnabled() const {
125    return randomVelocityEnabled;
126  }
127
128  void Squidbox::setRandomVelocityEnabled(bool enabled) {
129    randomVelocityEnabled = enabled;
130  }
131
132  void Squidbox::toggleRandomVelocity() {
133    randomVelocityEnabled = !randomVelocityEnabled;
134  }
135
136  uint8_t Squidbox::getNoteOnVelocity() {
137    if (randomVelocityEnabled) {
138      return random(randomVelocityMin, randomVelocityMax + 1);
139    }
140    return fixedVelocity;
141  }
142
143  uint8_t Squidbox::getFixedVelocity() const {
144    return fixedVelocity;
145  }
146
147  void Squidbox::setFixedVelocity(uint8_t velocity) {
148    fixedVelocity = velocity;
149  }
150
151  void Squidbox::setRandomVelocityRange(uint8_t minVelocity,
152                                        uint8_t maxVelocity) {
153    randomVelocityMin = minVelocity;
154    randomVelocityMax = maxVelocity;
155  }
156
157  uint8_t Squidbox::getRandomVelocityMin() const {
158    return randomVelocityMin;
159  }
160
161  uint8_t Squidbox::getRandomVelocityMax() const {
162    return randomVelocityMax;
163  }
```

We also created a function:

getNoteOnVelocity();

This function:
returns a fixed value when randomization is OFF
returns a random value when randomization is ON

3. **Updated All Scenes to Use Dynamic Velocity**

We updated all relevant scene files to replace hardcoded velocity values:

Before:
sendNoteOn(note, 127, 0);

After:
sendNoteOn(note, squidbox->getNoteOnVelocity(), 0);

```
void ChordScene::playChord(int index, bool on) {
  // Get the current scale, root note, and chord type from the menu items
  Scale *scale = scaleMenuItem->getScale();
  Note root = rootMenuItem->getRootNote();
  ChordType *chordType = chordTypeMenuItem->getChordType();

  // Get the notes in the chord
  int *notes = scale->getNotesFromChord(root, index, chordType);

  // Play or stop playing the notes
  for (int i = 0; i < chordType->numNotes; i++) {
    keyboard->setKeyDown(notes[i], on);
    if (on) { // If the chord should be played, play the note
      squidbox->getMidiController()->sendNoteOn(notes[i], squidbox->getNoteOnVelocity(), 0); //changed to have a randomized velocity if ena
    } else { // If the chord should be stopped, stop playing the note
      squidbox->getMidiController()->sendNoteOff(notes[i], 127, 0);
    }
  }
}
```

This change was applied to:

- NoteScene.cpp
- ChordScene.cpp
- DrumScene.cpp
- CustomScene.cpp

4**. Created a Menu Toggle**
We implemented a new menu item:
RandomVelocityMenuItem

This allows users to switch between:
Velocity: Fixed
Velocity: Random

The toggle is controlled using the device knob.

5. **Integrated Menu Into Scenes**

We added the new menu item into all playable scenes so users can toggle velocity behavior in real time.

```
4
5   ChordScene::ChordScene(Squidbox *squidbox) : Scene(squidbox, nullptr) {
6     // Create menu items for root note, scale, chord type, and random velocity
7     rootMenuItem = new RootNoteMenuItem();
8     scaleMenuItem = new ScaleMenuItem();
9     chordTypeMenuItem = new ChordTypeMenuItem();
10    randomVelocityMenuItem = new RandomVelocityMenuItem(squidbox);
11
12    // Create an array of menu items
13    MenuItem **menuItems = new MenuItem *[4];
14    menuItems[0] = rootMenuItem;
15    menuItems[1] = scaleMenuItem;
16    menuItems[2] = chordTypeMenuItem;
17    menuItems[3] = randomVelocityMenuItem;
18
19    // Create a new menu with the menu items
20    menu = new Menu("Chords", 4, menuItems, MAIN_SCENE);
21
22    // Create a new keyboard
23    keyboard = new Keyboard(squidbox);
24  }
25
```

Example (ChordScene):

menuItems[3] = randomVelocityMenuItem;